COMP 551 A1 Report

Altan Azimli, Laurel Johannson, Leonardo Martinez

September 30th 2025

1 Abstract

In this project we investigated the performance of two machine learning models on two open-source benchmark datasets from the UCI Machine Learning Repository. For a fully batched, 80/20 training/testing split, stochastic logistic regression to predict the cancer diagnosis of the Breast Cancer Wisconsin (Diagnostic) data set we achieved 94.29% accuracy in predicting the diagnosis of the testing set, while linear regression of motor scores from the Parkinson's Telemonitoring data set was able to fit the testing data with an R^2 of 0.869.

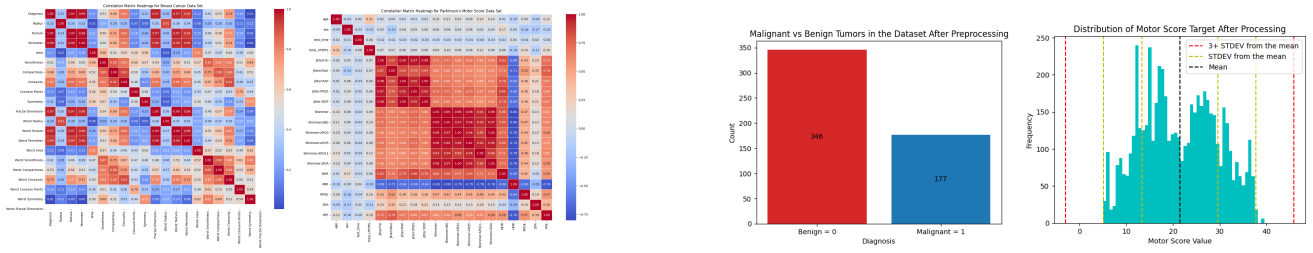
2 Introduction

The goal of this assignment is to use the Parkinson's Telemonitoring [1] and Breast Cancer Wisconsin (Diagnostic) [2] datasets from the open-source UCI Machine Learning Repository to implement linear and logistic regression algorithms. The Parkinson's Telemonitoring data was used by its originator Tsanas et al. to create a model to predict a standard score of disease progression using only voice and demographic data [3]. The Breast Cancer data set was orignally used by Street et al. to develop a model to diagnose tumours in the University of Wisconsin hospitals and represented the most accurate model at the time of publication [4]. As an exploratory analysis of the models' implementation, we performed several experiments comparing different configurations of the model parameters. For both models, the increase in performance associated with increasing the proportion on training data plateaued and smaller batch sizes were associated with decreased epochs to fit the data. In our tests, higher learning rates improved the runtime for linear SGD and improved accuracy for logistic SGD.

3 Datasets

The breast cancer data set originally contained 599 instances of 32 attributes - a numeric patient ID, a categorical diagnosis target variable, and the average, standard error, and maximum recorded measurement for 10 characteristics of images of a FNA for a breast mass. For our model, he standard error features were dropped because they describe the measurement rather than the image. Originally, the target diagnosis variable was encoded as 'M' for malignant or 'B' for benign, but for the purposes of logistic regression, this was changed to 1 and 0, respectively.

The Parkinson's data set was larger, containing 5875 instances of 19 features. The majority of the features describe measures of the voice samples from the participants with other variables describing demographic characteristics of the participants. The target variable for linear regression was the motor UPDRS feature, which was used as a measure of the progression of the patient's disease. Despite there being over 5000 instances in this data set, only 42 patients make up those instances.



(a) Correlation Matrices for the Datasets

(b) Distributions of Target Variables After Processing

Figure 1: Exploratory Analysis of the Datasets

For both data sets, the ID/subject number features were dropped and correlation analyses were performed. (figure X and X). For our models, we dropped features that were highly correlated (>0.9) with each other. For example, the perimeter

and area features were dropped from the Breast Cancer data set because they were highly correlated with the radius. We also processed the data by calculating the z score for each instance of a feature to exclude outliers. Assuming standard distributions for the features, only 0.2% of instances would be expected to be outside 3 standard deviations of the mean, so we felt comfortable excluding instances with z scores greater than 3 was sufficiently conservative. The features were also scaled for the purposes of reporting weights as this ensured the magnitude of the weights would be more consistent without impacting the performance of the models.

The use of open-source medical data sets to train AI models present several ethical issues. Weiner et al. (2025) discuss five critical ethical concerns for AI in healthcare, including patient consent and confidentiality [5]. These studies were performed in 2010 and 1993, over 15 and 30 years ago, respectively. Patients who consented to allowing their medical data to be published as part of these studies likely could not have predicted the extent to which AI models and LLMs would be trained on their data - calling into question whether their consent for their data to be used still stands. As highlighted by Weiner et al. (2025), the ability to revoke consent is also an issue with healthcare data sets and AI. There is a tension between the right of patients to opt-out at any time and the possibility that models have s already applied the data to learning algorithms [5].

4 Results

4.1 Fully-Batched Model Performance

Linear Regression

Training Set Performance:
Mean Squared Error: 6.465
Root Mean Squared Error: 2.543
R^2 Score: 0.890

Test Set Performance:
Mean Squared Error: 6.849
Root Mean Squared Error: 2.617
R^2 Score: 0.869

The linear regression model demonstrates strong predictive performance with an R^2 of 0.890 in the training set and 0.869 on the test set. The similar R^2 scores between training (0.890) and test (0.869) sets suggests that the model is not overfitting and generalizes appropriately to unseen data. The relatively low RMSE values indicate that predictions deviate from actual values by approximately 2.6 units on both the training set the test set. These results are good, which is to be expected since one of the dataset's targets total_UPDRS was included as a feature since excluding it yields terrible results. We included a command-line flag –true for both linear regression scripts to show what the performance would be like without including the target. In fact, we get an RMSE of 10.29 which is worse than the feature standard deviation of 8.13. This means that just thaking the expected value as our prediction would perform better than our model.
-

Logistic Regression

Training Set Performance:
Correct Classifications: $\frac{408}{418}$
Accuracy: 97.61%

Test Set Performance:
Correct Classifications: $\frac{99}{105}$
Accuracy: 94.29%

Unlike the linear regression model, the logistic regression model shows very good performance without having to include a target in the feature set. The accuracy on the test set is close to the accuracy on the training set and are both above 94%

4.2 Feature Weights from Fully-Batched Models

Linear Regression Final Weights:
age: -0.03135403260601891
sex: -0.14994192193379874
test_time: -0.04324162917263555
total_UPDRS: 7.263103911273387
Jitter(%): 1.8101184345900503
Jitter(Abs): -2.5035609710425577
Shimmer: 0.16488099270974654
NHR: -0.19392260878634038
HNR: -0.020652542179557587
RPDE: -0.16985463278116897
DFA: 0.18571079720582828
PPE: 0.5997956731617644

Note that the feature with the most weight is by far total_UPDRS since it is a target that was included in the feature set. After it Jitter(%) and Jitter(Abs) have the most weight in that order. They are both measures of jitter, which is a crucial voice measure used to qualify the instability or irregularity in a person's speech. Vocal instability is a well-documented symptom of Parkinson's, as the disease affects motor control including the muscles involved in speech production. These are the 3 only features whose weight has an absolute value above 1. Age and sex have relatively small weights, suggesting that motor symptom severity at any given time is more strongly related to vocal biomarkers and overall disease state.

Logistic Regression Final Weights:
radius1: 5.597292766017369
texture1: 1.7737097706383298
smoothness1: 1.6094668241182013
compactness1: -2.5375368168772754
concavity1: 2.069781151708283
symmetry1: 0.2569739083909749
fractal_dimension1: -1.0190702097486717
compactness3: 1.3890098990321156
smoothness3: -0.20590964347673257
area3: 1.0388192869569326
texture3: 0.20058354930717537
radius3: 1.328333305329933

The mean radius (radius1) has the strongest positive weight, indicating that larger cell nuclei are strongly associated with malignancy. This aligns with the medical knowledge that cancerous cells often have enlarged nuclei. The features that measure the complexity of the shape (concavity1, compactness1, texture1) have substantial weights. Malignant tumors typically exhibit more irregular cell shapes and textures compared to benign growths.

4.3 Training Data Split vs. Model Performance

For both fully-batched logistic regression and analytical linear regression models, 100 trials of increasing testing/training splits were performed and the average result and standard error for this average are reported in the figure below. As the proportion

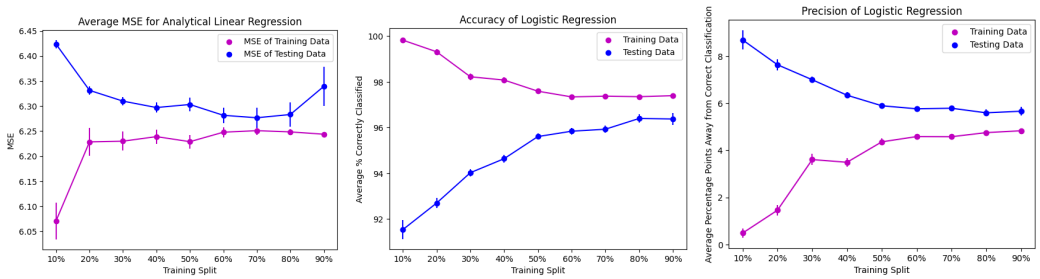


Figure 2: Model Performance vs. Training Split

of training data in the linear model increased, the MSE of the training data increased, but began to plateau around 60/40% training/testing data split. At the same time, the average standard error also shrank as the split increased, indicating less variability in the model weights as more training data was used. The testing data followed the opposite pattern, with the average MSE decreasing until around the 60/40% training/testing data split where the MSE began to plateau and the standard error began to increase. At all splits, the average MSE on the training data was lower than that of the testing data. This result suggests that beyond a 60/40% split, there is a decreasing rate of improvement in the prediction capability of the model.

For the logistic model, as the proportion of training data increased, the accuracy of the model on the training data increased while the precision decreased. Similar to the linear model, the testing data followed the opposite pattern. For this model, the results began to plateau around a 50/50% training/testing data split, suggesting that beyond this split, there is a decreasing rate of improvement in the prediction capability of the model.

4.4 Mini-Batch Size vs. Model Performance

The following tests were done using our scripts by doing [model].py –batch [value]
The percentage of the train set to be used as batch size is provided as a command-line argument
This is done the deafault 80/20 split and with the default respective learning rates of $\alpha = 1$ for linear and $\alpha = 10$ for logistic
Note that we are comparing the R^2 of the test sets and convergence speed will be measured by epochs and actual computation time. An epoch is one complete pass through the entire training dataset.

$$\text{Number of epochs} = \frac{\text{iterations} \times \text{batch size}}{\text{total training samples}}$$

Linear Regression

At a batch size of 1% of the training: $R^2 = 0.885$ Terminated after 100000 iterations (993.45 epochs) in 8.87 seconds
At a batch size of 10% of the training: $R^2 = 0.887$ Terminated after 100000 iterations (9979.68 epochs) in 9.85 seconds
At a batch size of 40% of the training: $R^2 = 0.889$ Terminated after 100000 iterations (39986.45 epochs) in 14.42 seconds
At a batch size of 60% of the training: $R^2 = 0.889$ Terminated after 100000 iterations (59990.97 epochs) in 16.92 seconds
Fully batched Stochastic Gradient Descent: $R^2 = 0.889$ Terminated after 100000 iterations (100000.00 epochs) in 22.08 seconds
We can see that the model performance is roughly the same regardless of the batch size but we see that we make net gains in minimizing the number of epochs to fit the model and in the actual compute time. Comparing the test with batch size of 1% to the fully batched one, we can observe that the compute time is nearly tripled in the latter and that we see 100×

more epochs. In other words, the fully batched model has seen 100× more data. This trend is consistent across the sampled batch sizes.

Logistic Regression

At a batch size of 1% of the training: 79.05% accuracy Terminated after 2 iterations (0.02 epochs) in 0.00 seconds
At a batch size of 10% of the training: 93.33% accuracy Terminated after 100000 iterations (9808.61 epochs) in 2.97 seconds
At a batch size of 40% of the training: 94.65% accuracy Terminated after 100000 iterations (39952.15 epochs) in 3.52 seconds
At a batch size of 60% of the training: 93.33% accuracy Terminated after 100000 iterations (59808.61 epochs) in 3.72 seconds
Fully batched Stochastic Gradient Descent: 93.33% accuracy Terminated after 100000 iterations (100000.00 epochs) in 4.23 seconds

We can see that the model performance dramatically improves with batch size 1% to 10% but afterwards performance remains somewhat consistent. Furthermore, once again, we see that we make net gains in minimizing the number of epochs to fit the model and in the actual compute time.

4.5 Learning Rates vs. Model Performance

Similar to the previous section, we explored differences in model performance based on a varying learning rate α. Using a mini-batched stochastic gradient descent model we will maintained the batch size at 10% and the train/test split at 80/20 constant across all tests for both linear and logistic regression.

Linear Regression
At a learning rate of 0.5: $R^2 = 0.889$ (test) — Terminated after 100,000 iterations (9979.68 epochs) in 6.45 seconds
At a learning rate of 1.0: $R^2 = 0.882$ (test) — Terminated after 100,000 iterations (9979.68 epochs) in 6.37 seconds
At a learning rate of 2.0: $R^2 = 0.875$ (test) — Terminated after 100,000 iterations (9979.68 epochs) in 6.32 seconds

The model's performance remains stable across different learning rates. We can see that with a larger alpha; the model reaches a solution faster but with a worst performance. As alpha increases, it takes less time for the model to run but with a decreasing R^2. This can be explained by the behaviour of the stochastic gradient descent: as alpha grows, the path to convergence exhibits more oscillatory behaviour which reduces accuracy. However, these larger steps taken by the path lead to a quicker run time, shown in Figure 4.

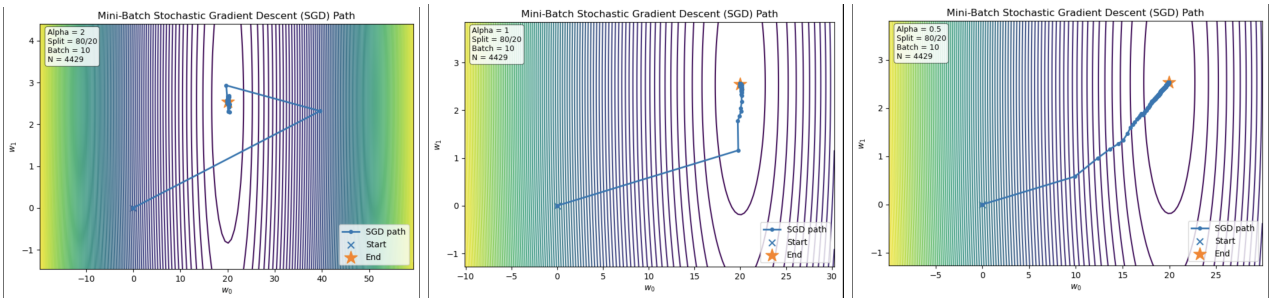


Figure 3: SGD Path Contour Maps for Linear Regression

Logistic Regression
At a learning rate of 5: Accuracy = 89.52% — Terminated after 99,999 iterations (9808.51 epochs) in 1.55 seconds
At a learning rate of 10:Accuracy = 90.48% — Terminated after 99,999 iterations (9808.51 epochs) in 1.54 seconds
At a learning rate of 15:Accuracy = 93.33% — Terminated after 99,999 iterations (9808.51 epochs) in 1.54 seconds

Here we observe a different behaviour in the model, although the run time remains near constant at 1.54 seconds, the accuracy increases. This due to the nature of the loss function for logistic regression. Its loss function is not a simple paraboloid like that of linear regression. Thus, larger alphas meaning bigger steps can lead to better convergence.

4.6 Analytical Solution vs. Mini-Batch Stochastic Gradient Descent for Linear Regression

The Analytical solution of a linear regression is solved by computing $(X^T X)^{-}1(X^T y)$ which gives the weights. We know that for very large and computationally costly data, SGD is faster. However as we see in this case, the analytical solution is quicker due to the dataset being relatively small. The following metrics were obtained from the analytical linear regression:
$R^2 = 0.869$ (test) — Terminated after 0.0006 seconds

$R^2 = 0.875$ (test) — Terminated after 6.32 seconds

The R^2 is nearly the same but the large improvement here is the efficiency of the model (0.0006 seconds vs. 6.32 seconds). This shows us that for computationally simple machine learning algorithms on a relatively small dataset, the analytical solution is the best.

5 Discussion and Conclusion

The goal of this project was to implement two machine learning methods on two benchmark, open-source health data sets and explore their performance under varying parameters. Both fully-batched models were adequate for describing and predicting values from their respective data sets, with linear regression fitting the training data with an R^2=0.890 and the testing data with R^2=0.869 and logistic regression correctly predicting 97.61% of diagnoses for the training set and 94.29% for the testing set. The Jitter(abs) feature had the highest weight in the linear regression model of the Parkinson's dataset, and the radius feature had the highest weight for the logistic regression model of the Breast Cancer data.

In our exploration of the models' performances, increasing the proportion of the dataset used for model training helped to more accurately predict the test data while decreasing the variability in model weights, but this effect plateaued for both linear and logistic regression. When implementing stochastic gradient descent, decreasing mini-batch size did not impact the accuracy of linear regression but improved runtimes. A similar effect on runtimes was observed for logistic regression. However, decreasing mini-batch sizes did not affect performance for logistic regression except at the smallest batch sizes, where it was noticeably deteriorated. Varying the learning rates also had different impacts on the two models - with higher learning rates improving runtimes for linear regression and accuracy for logistic regression. Finally, the SGD and analytical implementations of linear regression had similar accuracies, but the efficiency was better for the analytical solution, owing to a relatively small sample size.

We found that the motor and total UDPRS score features in the Parkinson's dataset were highly correlated. However, without including the motor UDPRS score as a feature the model did not perform well, this leaves room for a comparison of these two features as targets in future study. Similarly, an exploration of the temporal dimension to these scores along with a more robust statistical exploration of the impact of the relatively low patient population (N=42) could help to improve the performance of linear regression for this dataset. We were able to achieve high accuracy and precision for the logistic regression model of the Breast Cancer dataset, but could not meet the 97% accuracy attained in the original paper [3]. In future experiments, using a k-fold validation approach to determining model weights and accuracy could help to make a more robust model.

6 Statement of Contributions

Laurel Johannson prepared the initial data processing, performed experiment 4.3 and wrote the related report sections as well as the abstract, introduction, and conclusion of the report. Leonardo Martinez's implementation of the models was used, including the command-line argument support for each model. He also provided the result printing functionality of each model (linear.py, sgdlinear.py, logistic.py sgdlogistic.py). Additionally, he wrote sections 4.1,4.2,4.4 of the report. Altan Azimli performed experiments 4.5 and 4.6 and wrote the relevant report sections.

7 References

[1] A. Tsanas and M. Little. "Parkinsons Telemonitoring," UCI Machine Learning Repository, 2009. [Online]. Available: https://doi.org/10.24432/C5ZS3N.
[2] W. Wolberg, O. Mangasarian, N. Street, and W. Street. "Breast Cancer Wisconsin (Diagnostic)," UCI Machine Learning Repository, 1993. [Online]. Available: https://doi.org/10.24432/C5DW2B.
[3] W. Street, W. Wolberg, and O. Mangasarian, Nuclear feature extraction for breast tumor diagnosis (IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology). SPIE, 1993.
[4] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Accurate Telemonitoring of Parkinson's Disease Progression by Noninvasive Speech Tests," IEEE Transactions on Biomedical Engineering, vol. 57, no. 4, pp. 884-893, 2010, doi: 10.1109/TBME.2009.2036000.
[5] E. B. Weiner, I. Dankwa-Mullan, W. A. Nelson, and S. Hassanpour, "Ethical challenges and evolving strategies in the integration of artificial intelligence into clinical practice," (in eng), PLOS Digit Health, vol. 4, no. 4, p. e0000810, Apr 2025, doi: 10.1371/journal.pdig.0000810.